Архитектура клиента

Определение местонахождения сервера

- доменное имя или IP-адрес сервера могут быть заданы в виде константы во время трансляции клиентской программы;
- клиентская программа может требовать у пользователя указывать имя сервера при ее вызове;
- информация о местонахождении сервера предоставляется из постоянного хранилища данных (файл, база данных);
- для поиска сервера используется отдельный протокол.

Алгоритм клиента TCP:

1. Найти IP-адрес и номер порта протокола сервера, с которым необходимо установить связь.
2. Распределить сокет.
3. Указать, что для соединения нужен произвольный, неиспользуемый порт протокола на локальном компьютере и позволить ПО TCP выбрать такой порт.
4. Подключить сокет к серверу.
5. Выполнить обмен данными с сервером по протоколу прикладного уровня.
6. Закрыть соединение.

Синтаксис

–сначала записывается имя, а вторым параметром – порт;

–имя и порт рассматриваются как один параметр, разделенный двоеточием.

- Gethostbyname
- getservbyname
- Getprotobyname
- sockaddr_in
- socket
- int socket(int domain, int type, int protocol) - дескриптор

```
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
char * host = "localhost";
char * service = "imap";
char * proto = "tcp";
struct sockaddr_in sin;
```

```
int mksock( char *host, char * service, char * proto,
   struct sockaddr_in *sin)
{
struct hostent *hptr;
struct servent *sptr;
struct protoent *pptr;
int sd=0, type;
memset( sin, 0, sizeof( *sin));
sin->sin_family = AF_INET;
if( hptr = gethostbyname( host))
memcpy( & sin->sin_addr, hptr->h_addr,
   hptr->h_length);
else
```

```
return -1;
if ( ! ( pptr = getprotobyname( proto)) )
return -1;
if( sptr = getservbyname( service, proto))
sin->sin_port = sptr->s_port;
else if( (sin->sin_port =
htons(( unsigned short) atoi (service))) == 0)
return -1;
if ( strcmp( proto, "udp") == 0)
type = SOCK_DGRAM;
```

```
else
type = SOCK_STREAM;
if ( (sd = socket( PF_INET, type, pptr->p_proto)) < 0)
{
perror( "Ошибка при распределении сокета");
return -1;
}
return sd;
}
```

4 задачи:

- проверяет, является ли сокет действительным и не был ли он подключен;
- заполняет поле адреса конечной точки в дескрипторе сокета (из 2-го параметра);
- выбирает локальный адрес в дескрипторе сокета, если он еще не задан;
- инициирует соединение TCP и возвращает результат в вызывающую программу.

Вызов connect инициирует соединение локального сокета с удаленным :

int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen)

- send
- recv

int send(int s, const void *msg, size_t len,intflags);

int recv(int s, void *buf, size_t len, int flags).

Для обхода этой проблемы во многих реализациях интерфейса сокетов реализована функция shutdown:

int shutdown (int sd, int how)

где how принимает значения SHUT_RD, SHUT_WR, SHUT_RDWR (0, 1, 2 соответственно, однако рекомендовано использовать символьные константы), для закрытия сокета в одном из направлений. Для слушающего сокет это выглядит как конец файла.

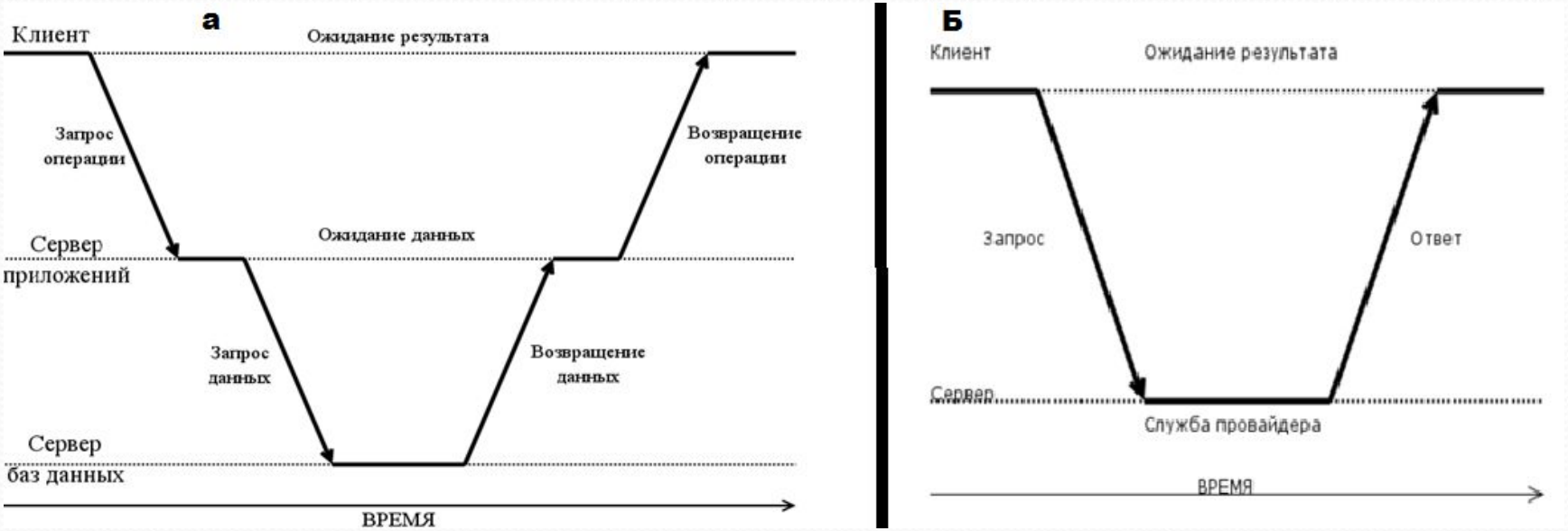
Алгоритм клиента UDP:

1. Найти IP-адрес и номер порта протокола сервера, с которым необходимо установить связь.
2. Распределить сокет.
3. Указать, что для соединения нужен произвольный, неиспользуемый порт протокола на локальном компьютере и позволить программному обеспечению UDP выбрать такой порт.
4. Указать сервер, на который должны передаваться сообщения.
5. Выполнить обмен данными с сервером по протоколу прикладного уровня.
6. Закрыть сокет.

Тест 2

- 1) назовите 3 уровня разделения приложений?
- 2) на каком уровне происходит управление клавиатурой, дисплеем?
- 3) протоколы TCP/IP обеспечивают **а)** многоранговую **б)** одноранговую связь?
- 4) какого распределения не существует: вертикального, горизонтального, диагонального, однорангового?
- 5) как называется распределение когда сервера может не быть вообще?
- 6) какая картинка изображает трехзвездную архитектуру клинет сервер?

- 1)Опишите простейшее взаимодействие 2х типов машин. (варианты архитектуры к-с)
- 2) 2 вощможные группы процессов?
- 3) Уровень пользовательского интерфейса обычно реализуется на **а)** клинетах **б)** серверах
- 4) на каком уровне реализуется логика обработки программ?
- 5)При физической трехзвездной архитектуре сервер приложения с точки зрения БД работает как: сервер, клиент, клиент и сервер одновременно?
- 6) какая картинка изображает **не** трехзвездную архитектуру клинет сервер?


а
Клиент
Ожидание результата
Запрос операции
Возвращение операции
Сервер приложений
Ожидание данных
Запрос данных
Возвращение данных
Сервер баз данных
ВРЕМЯ
Б
Клиент
Ожидание результата
Запрос
Ответ
Сервер
Служба провайдера
ВРЕМЯ